

Annexes

Sommaire des Annexes

SOMMAIRE DES ANNEXES	39
ANNEXE 1 : LISTE DES RESSOURCES PUBLIQUES UTILISEES.....	40
ANNEXE 2 : ETUDE UML DU PROJET	41
DIAGRAMME DES CAS D'UTILISATION	41
DIAGRAMME DES CLASSES : DICTIONNAIRE	42
DIAGRAMME DE CLASSES : ADMINISTRATION	43
DIAGRAMME DES CLASSES : CONTROLE	44
ANNEXE 3 : FICHIERS MANIPULES.....	45
EXEMPLE DE FICHIER XML : MODELE DEFAULT	45
EXEMPLE DE SCHEMA DE FICHIER : SCHEMA CAPTEURS.....	46
EXEMPLE D'UN FICHIER XSL : FICHIER UTILISE POUR L'AFFICHAGE DES DRAPEAUX	47
EXEMPLE DE FICHIER JSP : AFFICHAGE DE LA RECHERCHE ET SELECTION SERVEURS	48
ANNEXE 4 : COMMENT FABRIQUE-T-ON L'ACIER ?	49
AGGLOMERATION	50
LA COCKERIE	51
LE HAUT FOURNEAU	51
L'ACIERIE.....	51
LA COULEE CONTINUE	52
LAMINOIR.....	52
ANNEXE 5 : ARTICLE DE JOURNAL	52

Annexe 1 : Liste des ressources publiques utilisées

Ouvrages

UML en action, Eyrolles

XML avec Java, Eyrolles

JAVA côté serveur, Eyrolles

Sites Web

<http://www.xmlfr.org>

<http://www.application-servers.com>

<http://xml.apache.org>

<http://jakarta.apache.org/tomcat>

<http://jakarta.apache.org/struts>

Documents Electroniques

XSLT and XPATH Quick Reference, Mulberry Technologies, Inc.

UML Reference Card, Allen I. Holub.

OpenSource Java : MVC2 with Struts, Improve.

OpenSource Java : Servlet, JSP and MVC with Tomcat, Improve.

Annexe 2 : Etude UML du projet

Diagramme des cas d'utilisation

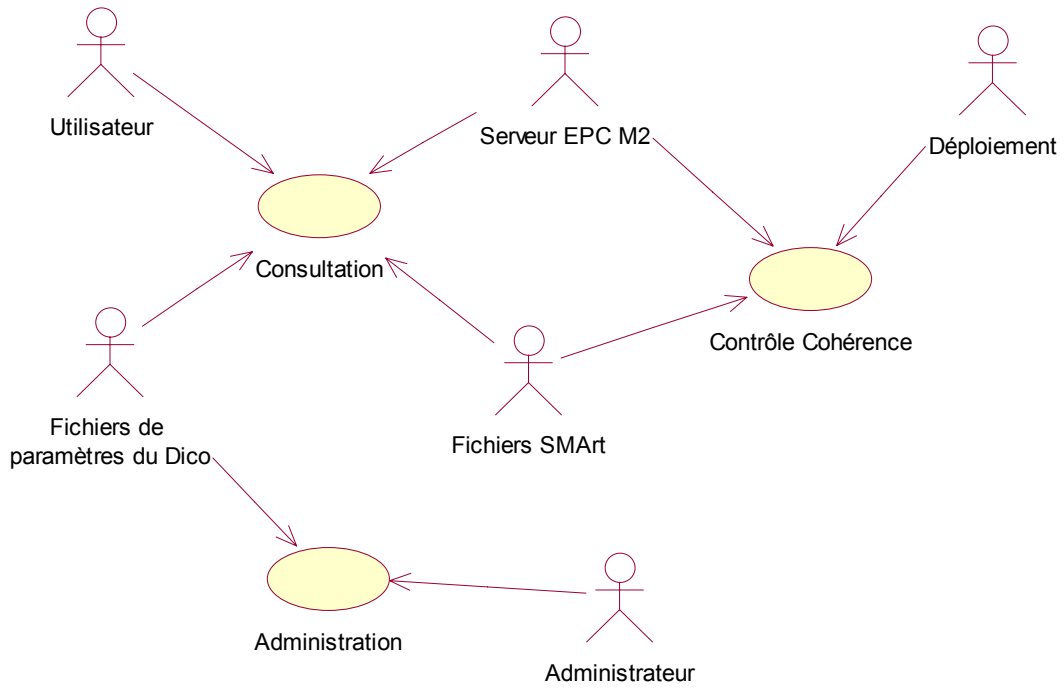


Diagramme des classes : Dictionnaire

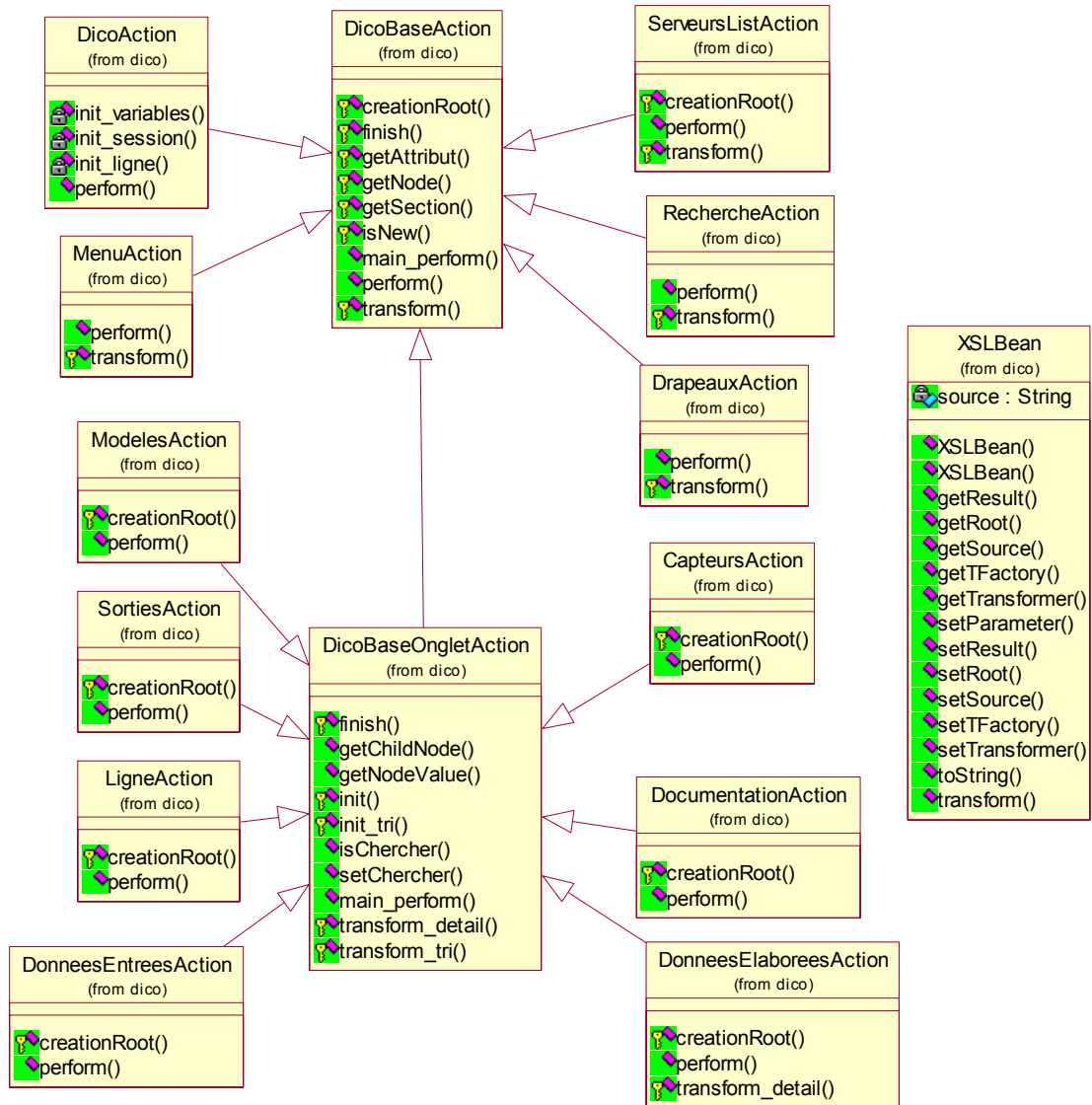


Diagramme de classes : Administration

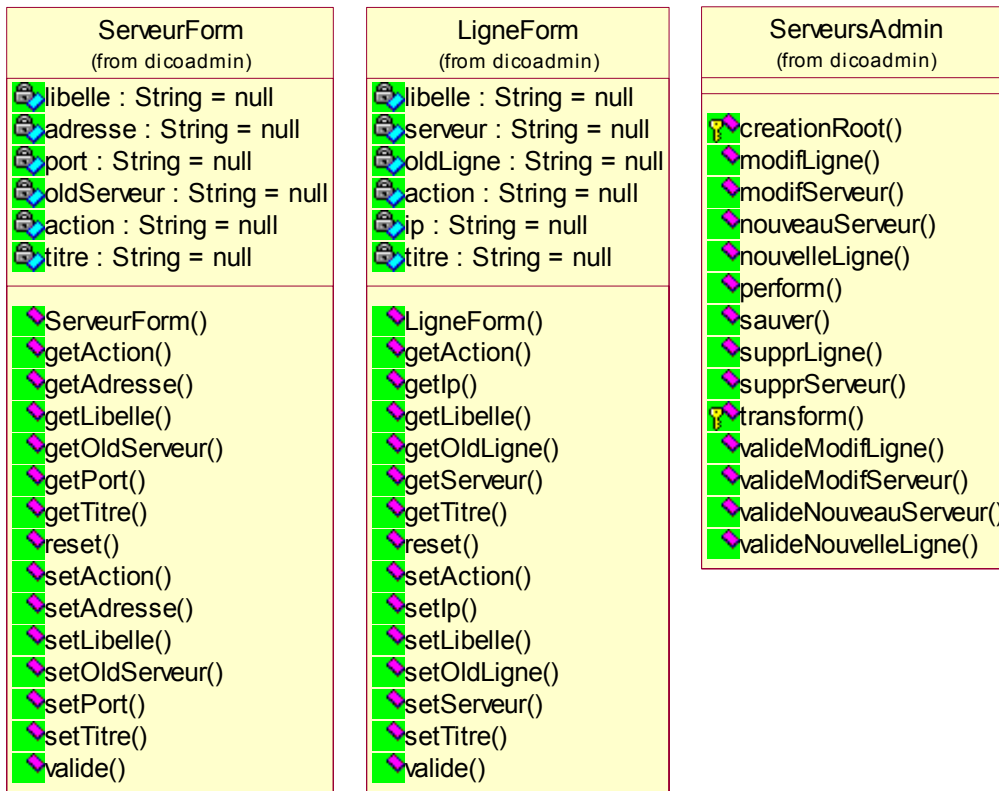
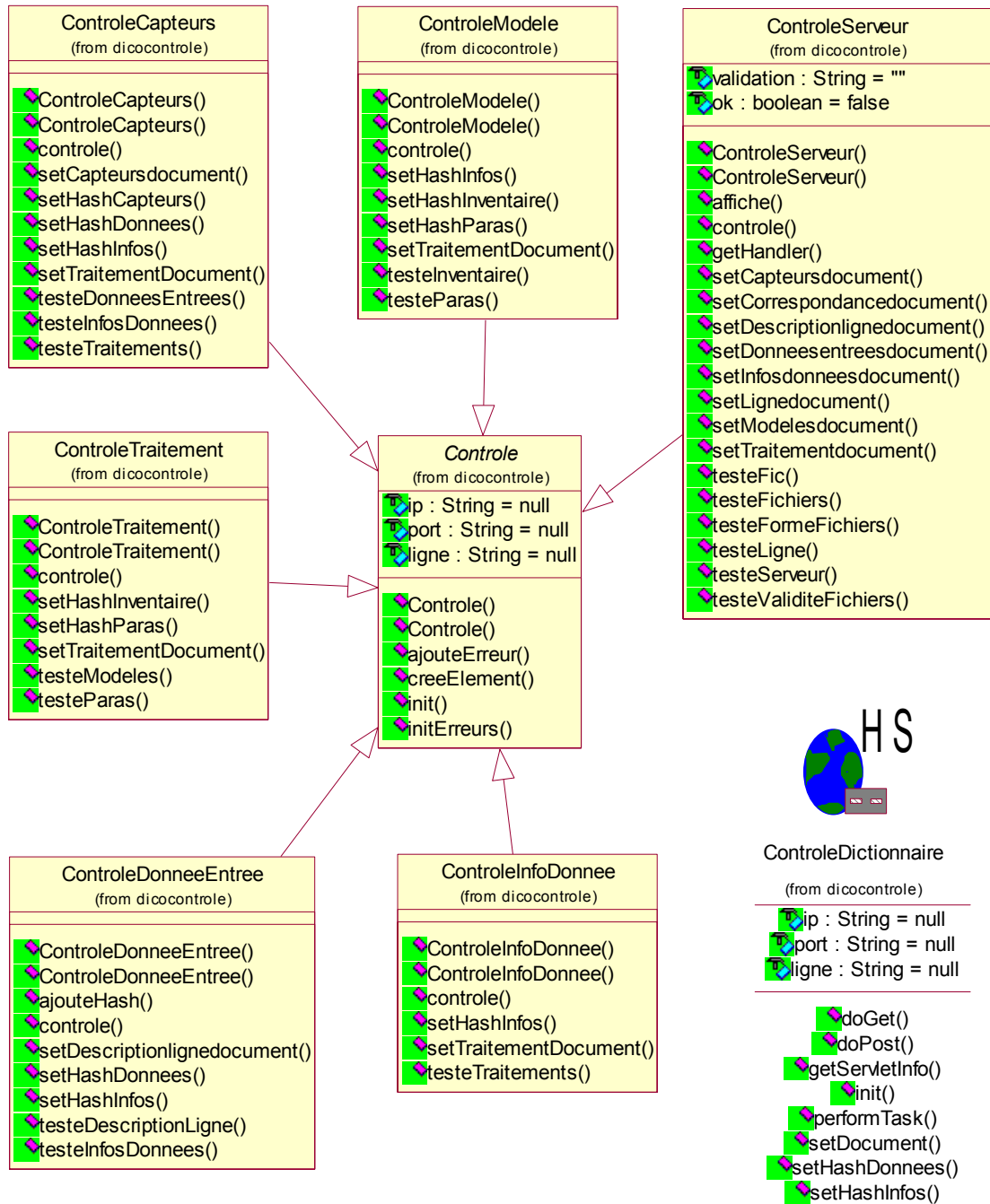


Diagramme des classes : Contrôle



Annexe 3 : Fichiers Manipulés

Exemple de fichier XML : Modèle Défaut

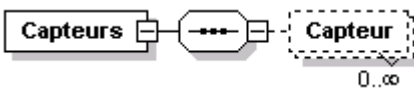
```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SMARt [
  <!ENTITY % HTMLlat1 PUBLIC "-//W3C//ENTITIES Latin1//EN//HTML" "http://web-dicoepc-
dev/xml/HTMLlat1x.ent">
  %HTMLlat1;
]>
<Modele>
  <Id>Ccp</Id>
  <Version>1.0.1</Version>
  <Auteur>Mathis G</Auteur>
  <Nom lang="fr">Calamine Cylindre pel&#233; Fos</Nom>
  <Nom lang="de">Calamine Cylindre pel&#233; Fos</Nom>
  <Nom lang="en">Calamine Cylindre pel&#233; Fos</Nom>
  <Nom lang="es">Calamine Cylindre pel&#233; Fos</Nom>
  <Type1>DEFAULT</Type1>
  <Type2>RB</Type2>
  <Description lang="fr">Modele default de detection des defauts de type "Calamine Cylindre Pel&#233;" pour le site
de Fos</Description>
  <Description lang="en">Modele default de detection des defauts de type "Calamine Cylindre Pel&#233;" pour le site
de Fos</Description>
  <Description lang="es">Modele default de detection des defauts de type "Calamine Cylindre Pel&#233;" pour le site
de Fos</Description>
  <Description lang="de">Modele default de detection des defauts de type "Calamine Cylindre Pel&#233;" pour le site
de Fos</Description>
  <Date>22/01/2001</Date>
  <Prototype>
    <In>para:TAC</In>
    <In>para:glissementF1</In>
    <In>para:glissementF2</In>
    <In>para:typeMontageTAC</In>
    <In>para:MbarF7</In>
    <In>para:NbCoilsF1</In>
    <In>para:Durete</In>
    <In>para:RedF2</In>
    <In>para:RedF1</In>
    <In>para:glissementF3</In>
    <In>para:epCoil2</In>
    <In>para:CAL_PU_F2</In>
    <In>para:MbarF1</In>
    <In>para:MbarF6</In>
    <In>para:TF1Moy5Coils</In>
    <In>para:CAL_PU_F1</In>
    <In>para:reducF1Moy5Coils</In>
    <In>para:aspectTAC</In>
    <In>para:Si</In>
    <In>para:TF1Moy5Coils</In>
    <In>para:densite_Ccp</In>
    <Out>para:p_densite_Ccp</Out>
  </Prototype>
  <ServMod>http://web-servmod-dev/accueil/princ.jsp?id=310-0</ServMod>
  <Ressource>
    <Code type="RB_NETICA">CCP22-01-01.dne</Code>
  </Ressource>
</Modele>

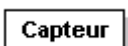
```

Exemple de Schéma de fichier : schéma Capteurs


element Capteurs

diagram	
children	Capteur
used by	element SMARt
source	<pre> <xsd:element name="Capteurs"> <xsd:complexType> <xsd:sequence> <xsd:element name="Capteur" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="CapteurType"> <xsd:attribute name="autoriseZero" type="xsd:string" use="optional"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </xsd:sequence> </xsd:complexType> </xsd:element> </pre>

element Capteurs/Capteur

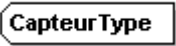
diagram																													
type	extension of CapteurType																												
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>xsd:ID</td> <td>required</td> <td></td> </tr> <tr> <td>type</td> <td>xsd:CDATA</td> <td>optional</td> <td></td> </tr> <tr> <td>min</td> <td>xsd:CDATA</td> <td>optional</td> <td></td> </tr> <tr> <td>max</td> <td>xsd:CDATA</td> <td>optional</td> <td></td> </tr> <tr> <td>zone</td> <td>xsd:CDATA</td> <td>optional</td> <td></td> </tr> <tr> <td>autoriseZero</td> <td>xsd:string</td> <td>optional</td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Value	id	xsd:ID	required		type	xsd:CDATA	optional		min	xsd:CDATA	optional		max	xsd:CDATA	optional		zone	xsd:CDATA	optional		autoriseZero	xsd:string	optional	
Name	Type	Use	Value																										
id	xsd:ID	required																											
type	xsd:CDATA	optional																											
min	xsd:CDATA	optional																											
max	xsd:CDATA	optional																											
zone	xsd:CDATA	optional																											
autoriseZero	xsd:string	optional																											
source	<pre> <xsd:element name="Capteur" minOccurs="0" maxOccurs="unbounded"> <xsd:complexType> <xsd:complexContent> <xsd:extension base="CapteurType"> <xsd:attribute name="autoriseZero" type="xsd:string" use="optional"/> </xsd:extension> </xsd:complexContent> </xsd:complexType> </xsd:element> </pre>																												

element SMARt

diagram	
children	Capteurs

source	<pre><xsd:element name="SMArt"> <xsd:complexType> <xsd:sequence> <xsd:element ref="Capteurs"/> </xsd:sequence> </xsd:complexType> </xsd:element></pre>
--------	--

complexType **CapteurType**

diagram				
used by	element <u>Capteurs/Capteur</u>			
attributes	Name	Type	Use	Value
	id	xsd:ID	Required	
	type	xsd:CDATA	optional	
	min	xsd:CDATA	optional	
	max	xsd:CDATA	optional	
	zone	xsd:CDATA	optional	
source	<pre><xsd:complexType name="CapteurType"> <xsd:attribute name="id" type="xsd:ID" use="required"/> <xsd:attribute name="type" type="xsd:CDATA" use="optional"/> <xsd:attribute name="min" type="xsd:CDATA" use="optional"/> <xsd:attribute name="max" type="xsd:CDATA" use="optional"/> <xsd:attribute name="zone" type="xsd:CDATA" use="optional"/> </xsd:complexType></pre>			

Exemple d'un fichier XSL : fichier utilisé pour l'affichage des drapeaux

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/TR/xhtml1/strict"
  version="1.0">

  <xsl:output method="html" indent="yes"/>
  <xsl:param name="lang"/>
  <xsl:param name="sel"/>

  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template >

  <xsl:template match="section">
    <table cellpadding="0" cellspacing="1" border="0" width="100%" id="latable" bgcolor="#EEEEEE">
      <tr>
```

```

    <xsl:choose>
      <xsl:when test="$sel">
        <xsl:apply-templates select="./item"/>
      </xsl:when>
      <xsl:otherwise>
        <td class="bordure" align="center"> Bienvenue sur le Dictionnaire EPC. Veuillez sélectionner un
serveur et une ligne.</td>
      </xsl:otherwise>
    </xsl:choose>
  </tr>
</table>
</xsl:template>

<xsl:template match="item">
  <xsl:apply-templates select="./value"/>
</xsl:template>

<xsl:template match="value">
  <xsl:if test="@lang=$lang">
    <td align="center" width="14%">
      <xsl:attribute name="class"><xsl:if test="../@nl = $sel">selectedmenu</xsl:if><xsl:if test="../@nl !=
$sel">bordure</xsl:if></xsl:attribute>
      <xsl:attribute name="id"><xsl:value-of select="../@nl"/></xsl:attribute>
      <xsl:element name="a">
        <xsl:attribute name="href"/><xsl:value-of select="../@nl"/></xsl:attribute>
        <xsl:attribute name="class"><xsl:if test="../@nl = $sel">selectedmenu</xsl:if><xsl:if test="../@nl !=
$sel">bordure</xsl:if></xsl:attribute>
        <xsl:attribute name="id"><xsl:value-of select="../@nl"/></xsl:attribute>
        <b><xsl:value-of select="."/></b>
      </xsl:element>
    </td>
  </xsl:if>
</xsl:template>
</xsl:stylesheet>

```

Exemple de fichier JSP : affichage de la recherche et sélection serveurs

```

<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%
  String SECTION = (String) session.getAttribute("sel");
  if (SECTION == null) SECTION = "";
%>
<head>
<html:base/>
<base target="_parent">
<script src="/js/choixServeur.js"></script>
<link rel="stylesheet" href="/css/dico.css" type="text/css">
<!-- Script pour afficher/cacher les infos de chaque serveur -->
<script language="javascript">
  function showhide(div) {
    if (document.all[div].style.display=="inline")
      {document.all[div].style.display = "none";}
    else {document.all[div].style.display = "inline";}
  }
</script>

```

```

<!-- Fin script afficher/cacher -->
</head>
<body>
<STYLE>
  A{color:#000000; text-decoration:none;}
  A:hover{text-decoration:underline;}
</STYLE>
<div style="position:absolute; top:80; left:45;">
  <font size="1"><a href="/dico.do?sel=deconnecter">Accueil</a>
</div>
<%
  if (!(SECTION.equals("")||SECTION.equals("deconnecter"))) {
%>
<div style="position:absolute; top:100; left:33;">
  <font size="1"><a target="gauche" href="#" onClick="showhide('selectmenu');">Liste des serveurs</a>
</div>
<div style="position:absolute; top:120; left:22;"></a>
  <font size="1"><a target="gauche" href="#" onClick="showhide('recherche');">Recherche</a>
</div>
<% }%>

<div id="selectmenu" style="display:none;">
  <bean:write name="serveursListResult" property="result" filter="false"/>
</div>
<br/>
<div id="recherche" style="display:none;">
  <bean:write name="rechercheResult" property="result" filter="false"/>
</div>
<br/>
<center>

</center>
<% if (SECTION.equals("")||SECTION.equals("deconnecter")) {
  out.println("<SCRIPT>showhide('selectmenu');</SCRIPT>");
}
%>
</body>
</html:html>

```

Annexe 4 : Comment fabrique-t-on l'acier ?

L'acier liquide est élaboré à partir du minerai (filrière fonte) ou à partir de ferrailles (filrière électrique). Ensuite, l'acier liquide est solidifié par moulage dans une machine de coulée continue. A la sortie, on obtient les DEMI-PRODUITS : des barres de section rectangulaire (brames) ou carrée (blooms ou billettes), qui sont les ébauches des formes finales. Enfin, les ébauches sont transformées en PRODUITS FINIS par laminage, dont certains subissent un traitement thermique. Plus de la moitié des tôles laminées à chaud sont relaminées à froid et éventuellement revêtues d'une protection anti-corrosion.

La fabrication de l'acier

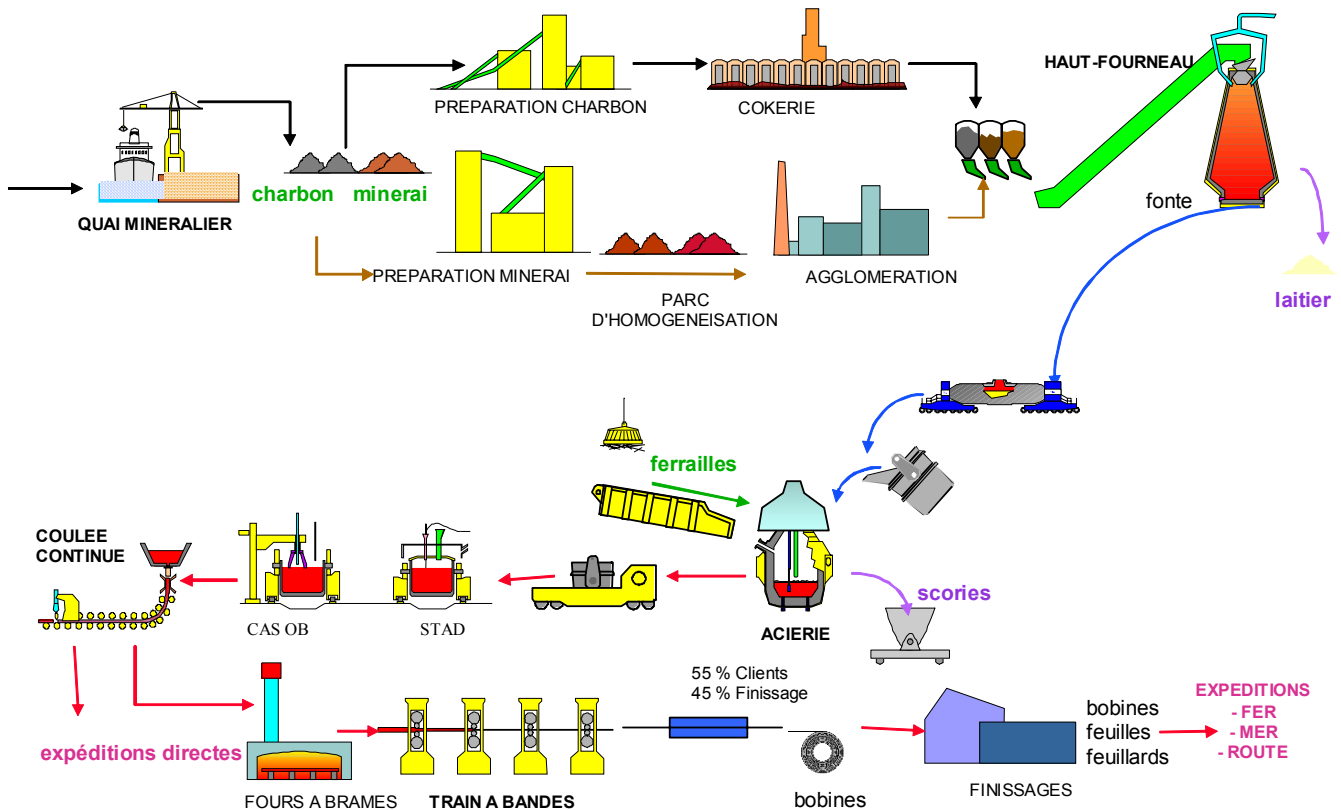
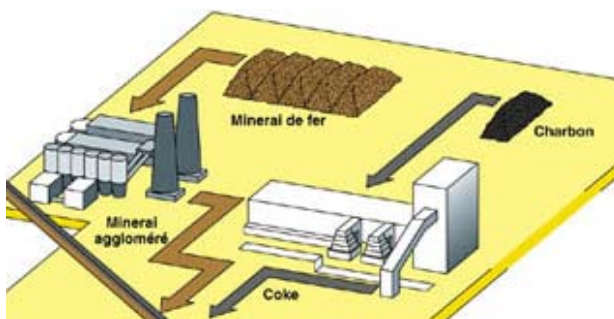


Figure 1 La fabrication de l'acier

Agglomération



Le minerai de fer est préparé, broyé et calibré en grains qui s'agglomèrent (= s'agglutinent) entre eux.

L'aggloméré obtenu est concassé puis chargé dans le haut fourneau avec du coke.

Le coke est un combustible puissant, résidu solide de la distillation de la houille (variété de charbon très riche en carbone).

La Cokerie

Le coke est un combustible obtenu par distillation (gazéification des composants indésirables) de la houille dans le four de la cokerie.

Le coke est du carbone presque pur doté d'une structure poreuse et résistante à l'écrasement.

En brûlant dans le haut fourneau, le coke apporte la chaleur nécessaire à la fusion du minerai et les gaz nécessaires à sa réduction.



Le Haut Fourneau



On extrait le fer de son minerai. Minerai et coke solides sont enfournés par le haut. L'air chaud (1200 °C) insufflé à la base provoque la combustion du coke (carbone presque pur). L'oxyde de carbone ainsi formé va "réduire" les oxydes de fer, c'est-à-dire leur prendre leur oxygène et, de ce fait, isoler le fer.

La chaleur dégagée par la combustion fait fondre fer et gangue en une masse liquide où la gangue, de densité moindre, flotte sur un mélange à base de fer, appelé "fonte".

Les résidus formés par la gangue fondue (laitiers) sont exploités par d'autres industries : construction de routes, cimenterie,...

L'aciérie

On y convertit la fonte en acier.

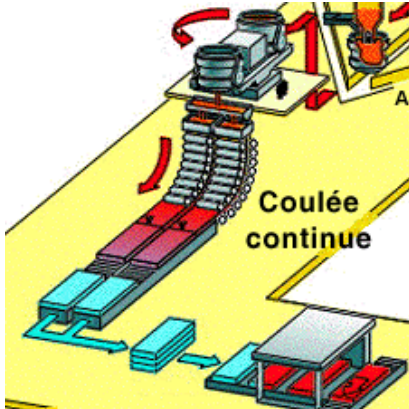
La fonte en fusion est versée sur un lit de ferraille.

On brûle les éléments indésirables (carbone et résidus) contenus dans la fonte en insufflant de l'oxygène pur.

On récupère les résidus (laitier d'aciérie).



On obtient de l'acier liquide "sauvage", qui est versé dans une poche. Il est appelé acier sauvage, car, à ce stade, il est encore imparfait.



La coulée Continue

Moulage d'ébauches (demi-produits)

Ici : moulage d'une brame. L'acier en fusion s'écoule en continu dans un moule sans fond. En traversant ce moule, il commence à se solidifier au contact des parois refroidies à l'eau.

Le métal moulé descend, guidé par un jeu de rouleaux, et

continue de se refroidir.

Arrivé à la sortie, il est solidifié à coeur. Il est immédiatement coupé aux longueurs voulues.

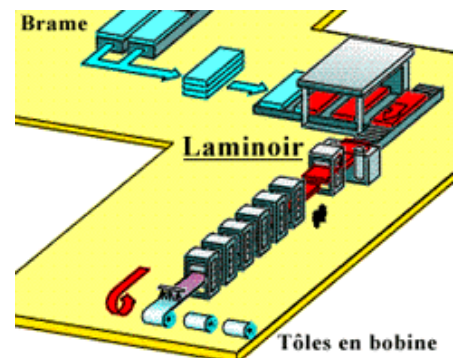
Laminoir

Des ébauches aux produits finis :

Ici : de la brame à la tôle.

La brame est réchauffée dans un four pour rendre le métal plus malléable, donc plus facile à étirer et à mettre en forme.

L'ébauche est ensuite amincie par écrasements progressifs entre les cylindres du laminoir.



Annexe 5 : Article de journal